

MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGGGG	FFFFFFFFFFFFFFFF	IIIIIIIIII	LLL
MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGGGG	FFFFFFFFFFFFFFFF	IIIIIIIIII	LLL
MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGGGG	FFFFFFFFFFFFFFFF	IIIIIIIIII	LLL
MMMMMM	MMMMMM	SSS	GGG	FFF	III	LLL
MMMMMM	MMMMMM	SSS	GGG	FFF	III	LLL
MMMMMM	MMMMMM	SSS	GGG	FFF	III	LLL
MMM	MMM	SSS	GGG	FFF	III	LLL
MMM	MMM	SSS	GGG	FFF	III	LLL
MMM	MMM	SSS	GGG	FFF	III	LLL
MMM	MMM	SSSSSSSSSS	GGG	FFFFFFFFFFFFFFFF	III	LLL
MMM	MMM	SSSSSSSSSS	GGG	FFFFFFFFFFFFFFFF	III	LLL
MMM	MMM	SSSSSSSSSS	GGG	FFFFFFFFFFFFFFFF	III	LLL
MMM	MMM	SSS	GGG	GGGGGGGGGG	III	LLL
MMM	MMM	SSS	GGG	GGGGGGGGGG	III	LLL
MMM	MMM	SSS	GGG	GGGGGGGGGG	III	LLL
MMM	MMM	SSS	GGG	GGG	III	LLL
MMM	MMM	SSS	GGG	GGG	III	LLL
MMM	MMM	SSS	GGG	GGG	III	LLL
MMM	MMM	SSS	GGG	GGG	III	LLL
MMM	MMM	SSS	GGG	GGG	III	LLL
MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGG	FFF	IIIIIIIIII	LLLLLLLLLLLLLLLL
MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGG	FFF	IIIIIIIIII	LLLLLLLLLLLLLLLL
MMM	MMM	SSSSSSSSSSSS	GGGGGGGGGG	FFF	IIIIIIIIII	LLLLLLLLLLLLLLLL

```

LL          IIIIII          SSSSSSSS
LL          IIIIII          SSSSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SSSSSS
LL          II             SSSSSS
LL          II             SS
LL          II             SS
LL          II             SS
LL          II             SS
LLLLLLLLLLLL IIIIII          SSSSSSSS
LLLLLLLLLLLL IIIIII          SSSSSSSS

```

```
1 0001 0 MODULE sdlgen ( IDENT = 'V04-000' ) =
2 0002 0
3 0003 1 BEGIN
4 0004 1
5 0005 1
6 0006 1 *****
7 0007 1 *
8 0008 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
9 0009 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
10 0010 1 * ALL RIGHTS RESERVED.
11 0011 1 *
12 0012 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
13 0013 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
14 0014 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
15 0015 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
16 0016 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
17 0017 1 * TRANSFERRED.
18 0018 1 *
19 0019 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
20 0020 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
21 0021 1 * CORPORATION.
22 0022 1 *
23 0023 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
24 0024 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
25 0025 1 *
26 0026 1 *
27 0027 1 *****
28 0028 1
29 0029 1
30 0030 1 ++
31 0031 1
32 0032 1 FACILITY:
33 0033 1 Message Compiler
34 0034 1
35 0035 1 ABSTRACT:
36 0036 1 The Message Compiler translates message definition language
37 0037 1 into object modules. This module is called by the parser
38 0038 1 to generate an SDL file.
39 0039 1
40 0040 1 ENVIRONMENT:
41 0041 1 VAX/VMS operating system, unprivileged user mode.
42 0042 1
43 0043 1 AUTHOR:
44 0044 1 Jim Teague, November 1981
45 0045 1
46 0046 1 Modified by:
47 0047 1
48 0048 1 V03-005 GJA0097 Greg Awdziewicz 10-Aug-1984
49 0049 1 - Change version # in listing title to v04-00.
50 0050 1
51 0051 1 V03-004 JWT0095 Jim Teague 08-Feb-1983
52 0052 1 Fix sdl generator to handle long symbols.
53 0053 1
54 0054 1 V03-003 JWT0048 Jim Teague 06-Aug-1982
55 0055 1 Adjust sdl output.
56 0056 1
57 0057 1 V03-002 BLS0168 Benn Schreiber 2-Apr-1982
```

```
58      0058 1 | Rename SDLGEN.REQ to SDLGENREQ.REQ
59      0059 1 |
60      0060 1 | V03-001 JWT0026      Jim Teague      23-Mar-1982
61      0061 1 | Make sdlgen flexible enough to accept different tags.
62      0062 1 |
63      0063 1 |
64      0064 1 | --
65      0065 1 |
66      0066 1 |
67      0067 1 | Include files
68      0068 1 |
69      0069 1 |
70      0070 1 | LIBRARY 'SYS$LIBRARY:STARLET';      ! VMS common definitions
71      0071 1 |
72      0072 1 | REQUIRE 'SRC$:MSG.REQ';      ! Message common definitions
73      0310 1 |
74      0311 1 | REQUIRE 'SRC$:SDLGENREQ.REQ';      ! Structure for comment blocks
75      0358 1 |
76      0359 1 | Table of Contents
77      0360 1 |
78      0361 1 |
79      0362 1 | FORWARD ROUTINE
80      0363 1 |     sdl_start_mod,      ! Beginnings of an SDL module
81      0364 1 |     sdl_define_constant, ! Defines message or literal constant
82      0365 1 |     sdl_end_mod,      ! Endings of SDL module
83      0366 1 |     sdl_comment,      ! Output a comment
84      0367 1 |     sdl_put_record;    ! Outputs a line to SDL file
85      0368 1 |
86      0369 1 |
87      0370 1 | Macros used within sdlgen...
88      0371 1 |
89      0372 1 | MACRO
90      0373 1 |
91      0374 1 | ! Create descriptor for a string
92      0375 1 |
93      M 0376 1 |     DESC ( string ) =
94      0377 1 |         %CHARCOUNT ( string ), UPLIT BYTE ( string )%,
95      0378 1 |
96      0379 1 | ! Concatenate 2 or more strings
97      0380 1 |
98      M 0381 1 |     CONCAT ( result, source ) [] =
99      M 0382 1 |         CH$MOVE ( .source, (source+4), .result[0] + .result[1]);
100     M 0383 1 |         result[0] = .result[0] + .source;
101     0384 1 |         CONCAT ( result, %REMAINING )%;
102     0385 1 |
103     0386 1 |
104     0387 1 | Literals
105     0388 1 |
106     0389 1 | LITERAL
107     0390 1 |     system_bit = 1,      ! /SYSTEM facility qualifier flag
108     0391 1 |     prefix_bit = 2,      ! /PREFIX facility qualifier flag
109     0392 1 |     macro_bit = 3,      ! /MACRO facility qualifier flag
110     0393 1 |     dectohex_switch = %X'FFFF', ! Value at which to switch from
111     0394 1 |                                     decimal to hexadecimal
112     0395 1 |     mod_offset = 1,      ! column for SDL stmt "MODULE"
113     0396 1 |     con_offset = 4,      ! column for SDL stmt "CONSTANT"
114     0397 1 |     equals_offset = 22,  ! column for SDL stmt "EQUALS"
```

```
.. 115      0398 1      pre_offset = 40,      ! column for SDL stmt 'PREFIX'
.. 116      0399 1      cname_offset = 7,      ! column for constant names
.. 117      0400 1      facil_offset = 8,      ! column for 'FACILITY'
.. 118      0401 1      line_size = 132,      ! Length of output record buffer
.. 119      0402 1      comment_skip = 5;      ! columns to skip for sameline comment
.. 120      0403 1
.. 121      0404 1      !
.. 122      0405 1      ! Storage definitions
.. 123      0406 1      !
.. 124      0407 1      ! OWN
.. 125      0408 1
.. 126      0409 1      facilnam_buf : VECTOR[10,BYTE],
.. 127      0410 1
.. 128      0411 1      facilnam_dsc : VECTOR[2] INITIAL (0, facilnam_buf),
.. 129      0412 1
.. 130      0413 1      ! pointer to head of SCB (Sdl Comment Block)
.. 131      0414 1      scb_head,
.. 132      0415 1
.. 133      0416 1      ! pointer to most recently created SCB
.. 134      0417 1      last_scb,
.. 135      0418 1
.. 136      0419 1      ! flag to indicate whether SCB's are saved for this module
.. 137      0420 1      scbs_saved:      BYTE INITIAL(false),
.. 138      0421 1
.. 139      0422 1      ! variable to keep track of current offset for SDL comment records
.. 140      0423 1      current_offset:      INITIAL ( 0 ),
.. 141      0424 1
.. 142      0425 1      ! module initialized flag
.. 143      0426 1      mod_init:      BYTE INITIAL(false),
.. 144      0427 1
.. 145      0428 1      ! flag to indicate that comments appear within constant definitions
.. 146      0429 1      comment_const:      BYTE INITIAL(false),
.. 147      0430 1
.. 148      0431 1      ! prefix buffer
.. 149      0432 1      prefix:      VECTOR[line_size,BYTE],
.. 150      0433 1
.. 151      0434 1      ! symbol name prefix descriptor
.. 152      0435 1      prefix_desc:      VECTOR [2] INITIAL (0, prefix),
.. 153      0436 1
.. 154      0437 1      ! output buffer
.. 155      0438 1      output_buffer:      VECTOR [line_size, BYTE],
.. 156      0439 1
.. 157      0440 1      ! descriptor of output buffer
.. 158      0441 1      output_desc:      VECTOR [2] INITIAL (0,output_buffer);
.. 159      0442 1
.. 160      0443 1      !
.. 161      0444 1      ! Strings for use in building SDL output file
.. 162      0445 1      !
.. 163      0446 1      ! OWN
.. 164      0447 1
.. 165      0448 1      xl      :      VECTOR [2]
.. 166      0449 1      INITIAL(DESC(%STRING(' EQUALS %X','!XL',' '))),
.. 167      0450 1      ul      :      VECTOR [2]
.. 168      0451 1      INITIAL(DESC(%STRING(' EQUALS ', '!UL',' ')));
.. 169      0452 1      BIND
.. 170      0453 1
.. 171      0454 1      blank      =      $descriptor( ' ' ),
```

```
: 172      0455 1      dbl_quote  =      $descriptor( '' ),
: 173      0456 1      def         =      $descriptor( 'DEF' ),
: 174      0457 1      start_module =      $descriptor( 'MODULE' ),
: 175      0458 1      const       =      $descriptor( 'CONSTANT' ),
: 176      0459 1      pre         =      $descriptor( 'PREFIX' ),
: 177      0460 1      dollar      =      $descriptor( '$' ),
: 178      0461 1      tag         =      $descriptor( 'TAG', '' ),
: 179      0462 1      comma       =      $descriptor( ',' ),
: 180      0463 1      semicolon    =      $descriptor( ';' ),
: 181      0464 1      endmodule    =      $descriptor( 'END MODULE;' ),
: 182      0465 1      facility    =      $descriptor( 'FACILITY' ),
: 183      0466 1      sdlcomment  =      $descriptor( '/*' );
: 184      0467 1
: 185      0468 1      !
: 186      0469 1      ! External storage
: 187      0470 1
: 188      0471 1      EXTERNAL
: 189      0472 1          facility_name : VECTOR,
: 190      0473 1          sdl_rab : BBLOCK,          ! SDL output file RAB
: 191      0474 1          sdl_fab : BBLOCK;          ! SDL output file FAB
: 192      0475 1
: 193      0476 1      !
: 194      0477 1      ! External routines
: 195      0478 1
: 196      0479 1      EXTERNAL ROUTINE
: 197      0480 1          syntax_error,          ! signal syntax error
: 198      0481 1          rms_error,             ! signal RMS-type error
: 199      0482 1          LIB$GET_VM: ADDRESSING_MODE( GENERAL ),
: 200      0483 1          LIB$FREE_VM: ADDRESSING_MODE( GENERAL );
```

```

: 202 0484 1 GLOBAL ROUTINE sdl_start_mod (facility_desc, facility_number,
: 203 0485 1 macro_desc, facility_flags ) =
: 204 0486 1
: 205 0487 1 ++
: 206 0488 1
: 207 0489 1
: 208 0490 1 This routine will (1)produce an SDL module definition,
: 209 0491 1 (2)dump comments (if any) which appeared prior to the
: 210 0492 1 .FACILITY definition in the MSG file, (3)start a constant
: 211 0493 1 list, and (4) define the facility number. The module name
: 212 0494 1 is determined by either /PREFIX or the facility name.
: 213 0495 1 The macro suffix is defined only if /MACRO is specified,
: 214 0496 1 and is 'DEF' otherwise. The constant prefix, is defined
: 215 0497 1 by /PREFIX or the facility name, depending on which is
: 216 0498 1 present. A '$' is added to the end of the constant prefix
: 217 0499 1 if the /SYSTEM qualifier is present. The tag field will
: 218 0500 1 always be null, resulting in a tag field of '_'.
: 219 0501 1
: 220 0502 1 Inputs:
: 221 0503 1
: 222 0504 1 facility_desc:
: 223 0505 1 address of descriptor of facility name.
: 224 0506 1 /PREFIX qualifier is used if present.
: 225 0507 1
: 226 0508 1 facility_number:
: 227 0509 1 facility number passed by value
: 228 0510 1
: 229 0511 1 macro_desc:
: 230 0512 1 address of descriptor of string
: 231 0513 1 specified by the /MACRO qualifier
: 232 0514 1
: 233 0515 1 facility_flags:
: 234 0516 1 bitvector indicating which facility
: 235 0517 1 qualifiers are present
: 236 0518 1
: 237 0519 1 Outputs:
: 238 0520 1 none
: 239 0521 1
: 240 0522 1
: 241 0523 1 --
: 242 0524 1
: 243 0525 2 BEGIN
: 244 0526 2
: 245 0527 2 MAP
: 246 0528 2 def : VECTOR,
: 247 0529 2 facility_desc : REF VECTOR,
: 248 0530 2 macro_desc : REF VECTOR,
: 249 0531 2 facility_flags: REF BITVECTOR;
: 250 0532 2
: 251 0533 2 LOCAL
: 252 0534 2 head_buf : VECTOR [line_size, BYTE],
: 253 0535 2 head_desc : VECTOR [2],
: 254 0536 2 buffer : VECTOR [line_size, BYTE],
: 255 0537 2 mod_desc : VECTOR [2],
: 256 0538 2 buffer_desc : VECTOR [2],
: 257 0539 2 faoout_buf : VECTOR [line_size, BYTE],
: 258 0540 2 faoout_desc : VECTOR [2],
```

```

259      0541 2      suffix_desc :      VECTOR [2];
260      0542 2
261      0543 2      mod_init = true;
262      0544 2
263      0545 2      facilnam_dsc[0] = .facility_name[0];
264      0546 2      CH$MOVE(.facility_name[0], .facility_name[1], .facilnam_dsc[1]);
265      0547 2
266      0548 2      |
267      0549 2      | Take whatever prefix we receive (facility name or explicit prefix)
268      0550 2      |
269      0551 2      prefix_desc [0] = .facility_desc [0];
270      0552 2      prefix_desc [1] = .facility_desc [1];
271      0553 2
272      0554 2      |
273      0555 2      | If prefix supplied, remove "_" from the specified prefix
274      0556 2      |
275      0557 2      IF .facility_flags [prefix_bit]
276      0558 2      THEN
277      0559 2          prefix_desc [0] = .prefix_desc [0] - 1
278      0560 2
279      0561 2      |
280      0562 2      |
281      0563 2      | If no prefix supplied, use prefix from facility name;
282      0564 2      | add "$" only if /SYSTEM specified
283      0565 2      |
284      0566 2      |
285      0567 2      ELSE
286      0568 2          IF .facility_flags [system_bit]
287      0569 2          THEN
288      0570 2              (CONCAT ( prefix_desc, dollar ));
289      0571 2
290      0572 2      IF .facility_flags [system_bit]
291      0573 2      THEN
292      0574 2          (CONCAT (facilnam_dsc, dollar));
293      0575 2
294      0576 2
295      0577 2      sdl_put_record ( UPLIT ( DESC ( ' ' )));
296      0578 2
297      0579 2      current_offset = mod_offset;
298      0580 2      buffer_desc [0] = mod_offset;
299      0581 2      buffer_desc [1] = buffer;
300      0582 2
301      0583 2      CH$FILL ( ' ', mod_offset, buffer);
302      0584 2
303      0585 2      mod_desc [0] = .prefix_desc [0] - 1;
304      0586 2      mod_desc [1] = .prefix_desc [1];
305      0587 2      suffix_desc [0] = 3;
306      0588 2      suffix_desc [1] = .def [1];
307      0589 2
308      0590 2      IF .facility_flags [macro_bit]
309      0591 2      THEN
310      0592 2          |
311      0593 2          | If /MACRO name is specified, use it for the module name after
312      0594 2          | first removing the $ (the $ is added below)
313      0595 2          | There will be no DEF suffix, either.
314      0596 2          |
315      0597 2      BEGIN

```

```
316      0598      suffix_desc [0] = 0;
317      0599      mod_desc [0] = .macro_desc [0] - 1;
318      0600      mod_desc [1] = .macro_desc [1] + 1;
319      0601      END;
320      0602
321      P 0603      CONCAT ( buffer_desc,
322      P 0604      start_module,
323      P 0605      dollar,
324      P 0606      mod_desc,
325      P 0607      suffix_desc,
326      0608      semicolon );
327      0609
328      0610      sdl_put_record ( buffer_desc );
329      0611      sdl_put_record ( UPLIT ( DESC ( ' ' ) ) );
330      0612
331      0613      :
332      0614      : Set up and output a header indicating that this file was created by
333      0615      : MESSAGE and the time created.
334      0616
335      0617      head_desc [0] = line_size;
336      0618      head_desc [1] = head_buf;
337      0619
338      0620      sdl_put_record( sdlcomment );
339      0621
340      P 0622      $FAO(UPLIT(DESC('/* This SDL File Generated by VAX-11 Message V04-00 on !%D')),
341      0623      head_desc, head_desc, 0);
342      0624
343      0625      sdl_put_record ( head_desc );
344      0626      sdl_put_record ( sdlcomment );
345      0627
346      0628      :
347      0629      : If comments were saved, dump them NOW
348      0630
349      0631      IF .scbs_saved
350      0632      THEN
351      0633      BEGIN
352      0634      LOCAL
353      0635      com_desc:      VECTOR [2],      ! Temporary comment descriptor
354      0636      CBLOCK:      REF BLOCK [,BYTE], ! For Sdl Comment Blocks
355      0637      LAST_CBLOCK: REF BLOCK [,BYTE]; ! For freeing vm
356      0638
357      0639      :
358      0640      : Set comment block ptr to first in list
359      0641      :
360      0642      CBLOCK = .scb_head;
361      0643
362      0644      :
363      0645      : As long as there is a forward link...
364      0646
365      0647      WHILE .CBLOCK NEQ 0 DO
366      0648
367      0649      BEGIN
368      0650      LOCAL
369      0651      status;
370      0652
371      0653      com_desc [0] = .CBLOCK [SCB$B_CLEN]; ! Set up temporary
372      0654      com_desc [1] = .CBLOCK [SCB$T_TXT]; ! descriptor
```

```
373 0655 4
374 0656 4      sdl_put_record ( com_desc );          ! Output comment line
375 0657 4
376 0658 4      LAST_CBLOCK = .CBLOCK;                  ! Save block address
377 0659 4
378 0660 4      CBLOCK = .CBLOCK [SCB$L_NXTSCB];          ! Move ptr forward
379 0661 4
380 0662 4      !
381 0663 4      ! Free vm for last comment block
382 0664 4      !
383 0665 5      IF NOT ( status = LIB$FREE_VM(%REF(SCB$C_LENGTH + .com_desc [0]),
384 0666 5          LAST_CBLOCK ) )
385 0667 4      THEN
386 0668 4          SIGNAL ( .status );
387 0669 4
388 0670 3      END;
389 0671 2  END;
390 0672 2
391 0673 2      current_offset = con_offset;
392 0674 2      buffer_desc[0] = con_offset;
393 0675 2      CH$FILE ( ' ', con_offset, buffer );
394 0676 2
395 P 0677 2      CONCAT ( buffer_desc,
396 0678 2          const );
397 0679 2
398 0680 2      sdl_put_record ( buffer_desc );
399 0681 2
400 0682 2      current_offset = cname_offset;
401 0683 2      buffer_desc [0] = facil_offset;
402 0684 2      CH$FILE ( ' ', pre_offset, buffer );
403 0685 2
404 P 0686 2      CONCAT ( buffer_desc,
405 P 0687 2          dbl_quote,
406 P 0688 2          facility,
407 0689 2          dbl_quote );
408 0690 2
409 0691 2      buffer_desc [0] = MAX ( equals_offset, .buffer_desc[0]);
410 0692 2
411 P 0693 2      CONCAT ( buffer_desc,
412 0694 2          ul );
413 0695 2
414 0696 2      buffer_desc [0] = MAX ( pre_offset, .buffer_desc[0] );
415 0697 2
416 P 0698 2      CONCAT ( buffer_desc,
417 P 0699 2          pre,
418 P 0700 2          dbl_quote,
419 P 0701 2          facilnam_desc,
420 P 0702 2          dbl_quote,
421 0703 2          tag );
422 0704 2
423 0705 2      faoout_desc [0] = line_size;
424 0706 2      faoout_desc [1] = faoout_buf;
425 0707 2
426 0708 2      $FAO ( buffer_desc, faoout_desc [0], faoout_desc, .facility_number);
427 0709 2
428 0710 2      sdl_put_record ( faoout_desc);
429 0711 2
```

```
: 430      0712 2 RETURN true;
: 431      0713 2
: 432      0714 1 END;
```

														.TITLE		SDLGEN					
														.IDENT		\V04-000\					
														.PSECT		\$SPLITS,NOWRT,NOEXE,2					
20	4C	58	21	58	25	20	53	4C	41	55	51	45	20	00000	P.AAA:	.ASCII	\	EQUALS	%X!XL	\	:
		20	4C	55	21	20	53	4C	41	55	51	45	20	0000E	P.AAB:	.ASCII	\	EQUALS	!UL	\	:
													20	0001A	P.AAD:	.ASCII	\				:
														0001B		.BLKB	1				:
													00000001	0001C	P.AAC:	.LONG	1				:
													00000000	00020		.ADDRESS	P.AAD				:
													22	00024	P.AAF:	.ASCII	\				:
														00025		.BLKB	3				:
													00000001	00028	P.AAE:	.LONG	1				:
													00000000	0002C		.ADDRESS	P.AAF				:
													46 45 44	00030	P.AAH:	.ASCII	\	DEF	\		:
														00033		.BLKB	1				:
													00000003	00034	P.AAG:	.LONG	3				:
													00000000	00038		.ADDRESS	P.AAH				:
													20 45 4C 55 44 4F 4D	0003C	P.AAJ:	.ASCII	\	MODULE	\		:
														00043		.BLKB	1				:
													00000007	00044	P.AAI:	.LONG	7				:
													00000000	00048		.ADDRESS	P.AAJ				:
													54 4E 41 54 53 4E 4F 43	0004C	P.AAL:	.ASCII	\	CONSTANT	\		:
													00000008	00054	P.AAK:	.LONG	8				:
													00000000	00058		.ADDRESS	P.AAL				:
													20 58 49 46 45 52 50	0005C	P.AAN:	.ASCII	\	PREFIX	\		:
														00063		.BLKB	1				:
													00000007	00064	P.AAM:	.LONG	7				:
													00000000	00068		.ADDRESS	P.AAN				:
													24	0006C	P.AAP:	.ASCII	\	\$	\		:
														0006D		.BLKB	3				:
													00000001	00070	P.AAO:	.LONG	1				:
													00000000	00074		.ADDRESS	P.AAP				:
													22 22 20 47 41 54 20	00078	P.AAR:	.ASCII	\	TAG	'""\		:
														0007F		.BLKB	1				:
													00000007	00080	P.AAQ:	.LONG	7				:
													00000000	00084		.ADDRESS	P.AAR				:
													2C	00088	P.AAT:	.ASCII	\	,	\		:
														00089		.BLKB	3				:
													00000001	0008C	P.AAS:	.LONG	1				:
													00000000	00090		.ADDRESS	P.AAT				:
													3B	00094	P.AAV:	.ASCII	\	;	\		:
														00095		.BLKB	3				:
													00000001	00098	P.AAU:	.LONG	1				:
													00000000	0009C		.ADDRESS	P.AAV				:
													3B 45 4C 55 44 4F 4D 5F 44 4E 45	000A0	P.AAX:	.ASCII	\	END_MODULE;	\		:
														000AB		.BLKB	1				:
													0000000B	000AC	P.AAW:	.LONG	11				:
													00000000	000B0		.ADDRESS	P.AAX				:
													59 54 49 4C 49 43 41 46	000B4	P.AAZ:	.ASCII	\	FACILITY	\		:
													00000008	000BC	P.AAY:	.LONG	8				:

6C 69 46 20 4C 44 53 20 73 69 68 54 20 2A 2F
20 79 62 20 64 65 74 61 72 65 6E 65 47 20 65
20 6E 6F 20 30 30 2D 34 30 56 20 65 67 61 73
44 25 21

```

00000000' 000C0 .ADDRESS P.AAZ
      2A 2F 000C4 P.ABB: .ASCII \/*\
      000C6 .BLKB 2
00000002' 000C8 P.ABA: .LONG 2
00000000' 000CC .ADDRESS P.ABB
      20 000D0 P.ABD: .ASCII \ \
      000D1 .BLKB 3
00000001' 000D4 P.ABC: .LONG 1
00000000' 000D8 .ADDRESS P.ABD
      20 000DC P.ABF: .ASCII \ \
      000DD .BLKB 3
00000001' 000E0 P.ABE: .LONG 1
00000000' 000E4 .ADDRESS P.ABF
      000E8 P.ABH: .ASCII \/* This SDL File Generated by VAX-11 Mes\
      000F7
      00106
      00110 .ASCII \sage V04-00 on !%D\
      0011F
      00122
0000003A' 00124 P.ABG: .BLKB 2
00000000' 00128 .LONG 58
      .ADDRESS P.ABH
      .PSECT $OWN$,NOEXE,2

00000 FACILNAM_BUF:
      .BLKB 10
00000A .BLKB 2
00000000 0000C FACILNAM_DSC:
      .LONG 0
00000000' 00010 .ADDRESS FACILNAM_BUF
      00014 SCB_HEAD:
      .BLKB 4
      00018 LAST_SCB:
      .BLKB 4
00 0001C SCBS_SAVED:
      .BYTE 0
      0001D .BLKB 3
00000000 00020 CURRENT_OFFSET:
      .LONG 0
00 00024 MOD_INIT:
      .BYTE 0
00 00025 COMMENT_CONST:
      .BYTE 0
      00026 .BLKB 2
      00028 PREFIX: .BLKB 132
00000000 000AC PREFIX_DESC:
      .LONG 0
00000000' 000B0 .ADDRESS PREFIX
      000B4 OUTPUT_BUFFER:
      .BLKB 132
00000000 00138 OUTPUT_DESC:
      .LONG 0
00000000' 0013C .ADDRESS OUTPUT_BUFFER
0000000E 00140 XL: .LONG 14
00000000' 00144 .ADDRESS P.AAA
0000000C 00148 UL: .LONG 12
00000000' 0014C .ADDRESS P.AAB

```

BLANK= P.AAC
DBL_QUOTE= P.AAE
DEF= P.AAG
START_MODULE= P.AAI
CONST= P.AAK
PRE= P.AAM
DOLLAR= P.AAO
TAG= P.AAQ
COMMA= P.AAS
SEMICOLON= P.AAU
ENDMODULE= P.AAW
FACILITY= P.AAY
SDLCOMMENT= P.ABA
.EXTRN FACILITY_NAME, SDL_RAB
.EXTRN SDL_FAB, SYNTAX_ERROR
.EXTRN RMS_ERROR, LIB\$GET_VM
.EXTRN LIB\$FREE_VM, SYS\$FAO

.PSECT \$CODE\$,NOWRT,2

OFFC 00000

.ENTRY SDL_START_MOD, Save R2,R3,R4,R5,R6,R7,R8,- : 0484
R9,R10,R11
MOVAB SYS\$FAO, R11
MOVAB SDL_PUT_RECORD, R10
MOVAB FACILNAM_DSC, R9
MOVAB DOLLAR, R8
MOVAB -452(SP), SP
MOVB #1, MOD_INIT : 0543
MOVL FACILITY_NAME, FACILNAM_DSC : 0545
MOVC3 FACILITY_NAME, @FACILITY_NAME+4, - : 0546
@FACILNAM_DSC+4
MOVL FACILITY_DESC, R0 : 0551
MOVQ (R0), PREFIX_DESC
BBC #2, @FACILITY_FLAGS, 1\$: 0557
DECL PREFIX_DESC : 0559
BRB 2\$
BBC #1, @FACILITY_FLAGS, 3\$: 0568
ADDL3 PREFIX_DESC+4, PREFIX_DESC, R0 : 0570
MOVC3 DOLLAR, @DOLLAR+4, (R0)
ADDL2 DOLLAR, PREFIX_DESC
BBC #1, @FACILITY_FLAGS, 3\$: 0572
ADDL3 FACILNAM_DSC+4, FACILNAM_DSC, R0 : 0574
MOVC3 DOLLAR, @DOLLAR+4, (R0)
ADDL2 DOLLAR, FACILNAM_DSC
PUSHAB P.ABC : 0577
CALLS #1, SDL_PUT_RECORD
MOVL #1, CURRENT_OFFSET : 0579
MOVL #1, BUFFER_DESC : 0580
MOVAB BUFFER, BUFFER_DESC+4 : 0581
MOVB #32, BUFFER : 0583
SUBL3 #1, PREFIX_DESC, MOD_DESC : 0585
MOVL PREFIX_DESC+4, MOD_DESC+4 : 0586
MOVL #3, SUFFIX_DESC : 0587
MOVL DEF+4, SUFFIX_DESC+4 : 0588
BBC #3, @FACILITY_FLAGS, 4\$: 0590
CLRL SUFFIX_DESC : 0598

			5B	00000000G	00	9E	00002	
			5A	0000V	CF	9E	00009	
			59	0000'	CF	9E	0000E	
			58	0000'	CF	9E	00013	
			5E	FE3C	CE	9E	00018	
		18	A9		01	90	0001D	
			69	0000G	CF	D0	00021	
04	B9	0000G	DF	0000G	CF	28	00026	
			50	04	AC	D0	0002F	
		00A0	C9		60	7D	00033	
	06	10	BC		02	E1	00038	
				00A0	C9	D7	0003D	
					17	11	00041	
	24	10	BC		01	E1	00043	1\$:
	50	00A0	C9	00A4	C9	C1	00048	
	60	04	B8		68	28	00050	
		00A0	C9		68	C0	00055	
	0D	10	BC		01	E1	0005A	2\$:
	50		69	04	A9	C1	0005F	
	60	04	B8		68	28	00064	
			69		68	C0	00069	
				64	A8	9F	0006C	3\$:
			6A		01	FB	0006F	
		14	A9		01	D0	00072	
		00A4	CE		01	D0	00076	
		00A8	CE	00B4	CE	9E	0007B	
		00B4	CE		20	90	00082	
00AC	CE	00A0	C9		01	C3	00087	
		00B0	CE	00A4	C9	D0	0008F	
		10	AE		03	D0	00096	
		14	AE	C8	A8	D0	0009A	
	14	10	BC		03	E1	0009F	
				10	AE	D4	000A4	

00AC	CE		50	0C	AC	D0	000A7	MOVL	MACRO DESC, R0	0599
00B0	CE		60		01	C3	000AB	SUBL3	#1, (R0), MOD DESC	0600
	CE	04	A0		01	C1	000B1	ADDL3	#1, 4(R0), MOD DESC+4	0608
	50	00A4	CE	00A8	CE	C1	000B8	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	D8	B8	D4	A8	28	000C0	MOVC3	START_MODULE, @START_MODULE+4, (R0)	
		00A4	CE	D4	A8	C0	000C6	ADDL2	START_MODULE, BUFFER_DESC	
	50	00A4	CE	00A8	CE	C1	000CC	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	04	B8		68	28	000D4	MOVC3	DOLLAR, @DOLLAR+4, (R0)	
		00A4	CE		68	C0	000D9	ADDL2	DOLLAR, BUFFER_DESC	
	50	00A4	CE	00A8	CE	C1	000DE	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	00B0	DE	00AC	CE	28	000E6	MOVC3	MOD_DESC, @MOD_DESC+4, (R0)	
		00A4	CE	00AC	CE	C0	000EE	ADDL2	MOD_DESC, BUFFER_DESC	
	50	00A4	CE	00A8	CE	C1	000F5	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	14	BE	10	AE	28	000FD	MOVC3	SUFFIX_DESC, @SUFFIX_DESC+4, (R0)	
		00A4	CE	10	AE	C0	00103	ADDL2	SUFFIX_DESC, BUFFER_DESC	
	50	00A4	CE	00A8	CE	C1	00109	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	2C	B8	28	A8	28	00111	MOVC3	SEMICOLON, @SEMICOLON+4, (R0)	
		00A4	CE	28	A8	C0	00117	ADDL2	SEMICOLON, BUFFER_DESC	
			6A	00A4	CE	9F	0011D	PUSHAB	BUFFER_DESC	0610
				70	A8	9F	00121	CALLS	#1, SDL_PUT_RECORD	0611
			6A		01	FB	00127	PUSHAB	P.ABE	
		FF74	CD	84	8F	9A	0012A	CALLS	#1, SDL_PUT_RECORD	0617
		FF78	CD	FF7C	CD	9E	00130	MOVZBL	#132, HEAD_DSC	0618
				58	A8	9F	00137	MOVAB	HEAD_BUF, HEAD_DSC+4	0620
			6A		01	FB	0013A	PUSHAB	SDLCOMMENT	
					7E	D4	0013D	CALLS	#1, SDL_PUT_RECORD	0623
				FF74	CD	9F	0013F	CLRL	-(SP)	
				FF74	CD	9F	00143	PUSHAB	HEAD_DSC	
				00B4	C8	9F	00147	PUSHAB	HEAD_DSC	
			6B		04	FB	0014B	PUSHAB	P.ABG	
				FF74	CD	9F	0014E	CALLS	#4, SYSSFAO	0625
			6A		01	FB	00152	PUSHAB	HEAD_DSC	
				58	A8	9F	00155	CALLS	#1, SDL_PUT_RECORD	0626
			6A		01	FB	00158	PUSHAB	SDLCOMMENT	
			40	10	A9	E9	0015B	CALLS	#1, SDL_PUT_RECORD	0631
			52	08	A9	D0	0015F	BLBC	SCBS SAVED, 6\$	0642
					52	D5	00163	MOVL	SCB_READ, CBLOCK	0647
					38	13	00165	TSTL	CBLOCK	
		08	AE	04	A2	9A	00167	BEQL	6\$	0653
		0C	AE	05	A2	9E	0016C	MOVZBL	4(CBLOCK), COM_DESC	0654
				08	A2	9F	00171	MOVAB	5(R2), COM_DESC+4	0656
			6A		01	FB	00174	PUSHAB	COM_DESC	
		04	AE		52	D0	00177	CALLS	#1, SDL_PUT_RECORD	0658
		52			62	D0	0017B	MOVL	CBLOCK, LAST_CBLOCK	0660
				04	AE	9F	0017E	MOVL	(CBLOCK), CBLOCK	0665
04	AE	0C	AE	04	05	C1	00181	PUSHAB	LAST_CBLOCK	
					05	AE	9F	ADDL3	#5, COM_DESC, 4(SP)	
		00000000G	00		02	FB	0018A	PUSHAB	4(SP)	
			CF		50	E8	00191	CALLS	#2, LIB\$FREE_VM	
					50	DD	00194	BLBS	STATUS, 5\$	0668
		00000000G	00		01	FB	00196	PUSHL	STATUS	
					C4	11	0019D	CALLS	#1, LIB\$SIGNAL	0647
		14	A9		04	D0	0019F	BRB	5\$	0673
		00A4	CE		04	D0	001A3	MOVL	#4, CURRENT_OFFSET	0674
		00B4	CE	20202020	8F	D0	001A8	MOVL	#4, BUFFER_DESC	0675
50		00A4	CE	00A8	CE	C1	001B1	MOVL	#538976288, BUFFER	0678
								ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	

28

60	E8	B8	E4	A8	28	001B9	MOV C3	CONST, @CONST+4, (R0)	:	
	00A4	CE	E4	A8	C0	001BF	ADD L2	CONST, BUFFER_DESC	:	
			00A4	CE	9F	001C5	PUSH AB	BUFFER_DESC	:	0680
		6A		01	FB	001C9	CALL S	#1, SDI_PUT_RECORD	:	
	14	A9		07	D0	001CC	MOVL	#7, CURRENT_OFFSET	:	0682
	00A4	CE		08	D0	001DC	MOVL	#8, BUFFER_DESC	:	0683
20		6E		00	2C	001D5	MOV C5	#0, (SP), #32, #40, BUFFER	:	0684
			00B4	CE		001DA			:	
		56	B8	A8	7D	001DD	MOV Q	DBL QUOTE, R6	:	0689
50	00A4	CE	00A8	CE	C1	001E1	ADD L5	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60		67		56	28	001E9	MOV C3	R6, (R7), (R0)	:	
	00A4	CE		56	C0	001ED	ADD L2	#6, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	001F2	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60	50	B8	4C	A8	28	001FA	MOV C3	FACILITY, @FACILITY+4, (R0)	:	
	00A4	CE	4C	A8	C0	00200	ADD L2	FACILITY, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00206	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60		67		56	28	0020E	MOV C3	R6, (R7), (R0)	:	
	00A4	CE		56	C0	00212	ADD L2	R6, BUFFER_DESC	:	
		50	00A4	CE	D0	00217	MOVL	BUFFER_DESC, R0	:	0691
		16		50	D1	0021C	CMPL	R0, #22	:	
				03	18	0021F	BGE Q	7\$	:	
		50		16	D0	00221	MOVL	#22, R0	:	
	00A4	CE		50	D0	00224	7\$: MOVL	R0, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00229	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	0694
60	0140	D9	013C	C9	28	00231	MOV C3	UL, @UL+4, (R0)	:	
	00A4	CE	013C	C9	C0	00239	ADD L2	UL, BUFFER_DESC	:	
		50	00A4	CE	D0	00240	MOVL	BUFFER_DESC, R0	:	0696
		28		50	D1	00245	CMPL	R0, #40	:	
				03	18	00248	BGE Q	8\$	:	
		50		28	D0	0024A	MOVL	#40, R0	:	
	00A4	CE		50	D0	0024D	8\$: MOVL	R0, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00252	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	0703
60	F8	B8	F4	A8	28	0025A	MOV C3	PRE, @PRE+4, (R0)	:	
	00A4	CE	F4	A8	C0	00260	ADD L2	PRE, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00266	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60		67		56	28	0026E	MOV C3	R6, (R7), (R0)	:	
	00A4	CE		56	C0	00272	ADD L2	R6, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00277	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60	04	B9		69	28	0027F	MOV C3	FACILNAM_DSC, @FACILNAM_DSC+4, (R0)	:	
	00A4	CE		69	C0	00284	ADD L2	FACILNAM_DSC, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	00289	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60		67		56	28	00291	MOV C3	R6, (R7), (R0)	:	
	00A4	CE		56	C0	00295	ADD L2	R6, BUFFER_DESC	:	
50	00A4	CE	00A8	CE	C1	0029A	ADD L3	BUFFER_DESC+4, BUFFER_DESC, R0	:	
60	14	B8	10	A8	28	002A2	MOV C3	TAG, @TAG+4, (R0)	:	
	00A4	CE	10	A8	C0	002A8	ADD L2	TAG, BUFFER_DESC	:	
	18	AE	84	8F	9A	002AE	MOV ZBL	#132, FAOOUT_DESC	:	0705
	1C	AE	20	AE	9E	002B3	MOV AB	FAOOUT_BUF, FAOOUT_DESC+4	:	0706
			08	AC	DD	002B8	PUSH L	FACILITY NUMBER	:	0708
			1C	AE	9F	002BB	PUSH AB	FAOOUT_DESC	:	
			20	AE	9F	002BE	PUSH AB	FAOOUT_DESC	:	
			00B0	CE	9F	002C1	PUSH AB	BUFFER_DESC	:	
		6B		04	FB	002C5	CALL S	#4, SYSSFAO	:	
			18	AE	9F	002C8	PUSH AB	FAOOUT_DESC	:	0710
		6A		01	FB	002CB	CALL S	#1, SDI_PUT_RECORD	:	
		50		01	D0	002CE	MOVL	#1, R0	:	0712
				04	00	002D1	RET		:	0714

SDLGEN
V04-000

C 16
16-Sep-1984 02:11:38
14-Sep-1984 12:46:23

VAX-11 Bliss-32 V4.0-742
[MSGFIL.SRC]SDLGEN.B32;1

Page 14
(2)

; Routine Size: 722 bytes, Routine Base: \$CODE\$ + 0000

```

: 434 0715 1 GLOBAL ROUTINE sdl_define_constant ( name_desc, value, msg_flag, tparse) =
: 435 0716 1
: 436 0717 1 |++
: 437 0718 1
: 438 0719 1 |       This routine outputs a constant definition as follows
: 439 0720 1
: 440 0721 1 |       symbol_name EQUALS symbol_value PREFIX prefix TAG ''
: 441 0722 1
: 442 0723 1 |       This routine is capable of handling message symbols or
: 443 0724 1 |       literals.  In the latter case, it checks to be sure that
: 444 0725 1 |       the literal name is consistent with the MODULE statement.
: 445 0726 1 |       If there are inconsistencies, a warning is issued and the
: 446 0727 1 |       literal is omitted.
: 447 0728 1
: 448 0729 1
: 449 0730 1 |       Inputs:
: 450 0731 1
: 451 0732 1 |       name_desc:
: 452 0733 1 |           address of descriptor of symbol name
: 453 0734 1
: 454 0735 1 |       value:
: 455 0736 1 |           symbol value
: 456 0737 1
: 457 0738 1 |       msg_flag:
: 458 0739 1 |           set if a message symbol and not a literal
: 459 0740 1
: 460 0741 1 |       tparse:
: 461 0742 1 |           address of tparse block, when present
: 462 0743 1
: 463 0744 1 |       Outputs:
: 464 0745 1
: 465 0746 1 |       none
: 466 0747 1
: 467 0748 1
: 468 0749 1 | --
: 469 0750 1
: 470 0751 2 BEGIN
: 471 0752 2
: 472 0753 2 MAP
: 473 0754 2     comma      :      VECTOR,
: 474 0755 2     blank      :      VECTOR,
: 475 0756 2     name_desc  :      REF VECTOR;
: 476 0757 2
: 477 0758 2 LOCAL
: 478 0759 2     pre_dsc      :      VECTOR [2],
: 479 0760 2     delim_desc :      VECTOR [2],
: 480 0761 2     buffer      :      VECTOR [line_size, BYTE],
: 481 0762 2     buffer_desc :      VECTOR [2],
: 482 0763 2     faoout_desc :      VECTOR [2],
: 483 0764 2     faoout_buf  :      VECTOR [line_size, BYTE],
: 484 0765 2     symbol_desc :      VECTOR [2];
: 485 0766 2
: 486 0767 2 |
: 487 0768 2 | Make sure that we're not outside a module def'n
: 488 0769 2
: 489 0770 2 IF NOT .mod_init
: 490 0771 2 THEN
```

```
491 0772 3 BEGIN
492 0773 3 syntax_error(.tparse, emsg(nofacil)) ;
493 0774 3 RETURN true;
494 0775 2 END;
495 0776 2
496 0777 2
497 0778 2 Store new symbol name locally
498 0779 2
499 0780 2 symbol_desc [0] = .name_desc [0];
500 0781 2 symbol_desc [1] = .name_desc [1];
501 0782 2 buffer_desc [1] = buffer;
502 0783 2
503 0784 2
504 0785 2 Delimiter is a comma
505 0786 2
506 0787 2 delim_desc [0] = 1;
507 0788 2 delim_desc [1] = .comma [1];
508 0789 2
509 0790 2
510 0791 2 Assume no special prefix
511 0792 2
512 0793 2 pre_dsc [0] = .prefix_desc [0];
513 0794 2 pre_dsc [1] = .prefix_desc [1];
514 0795 2
515 0796 2
516 0797 2 Check to see if it's a message def'n or not
517 0798 2
518 0799 2 IF NOT .msg_flag
519 0800 2 THEN
520 0801 2 BEGIN
521 0802 2 LOCAL
522 0803 2 checklen;
523 0804 2
524 0805 2
525 0806 2 Compare prefix and first part of symbol all the way up to the
526 0807 2 first underscore
527 0808 2
528 0809 2 IF (checklen = (CH$FIND_CH( .symbol_desc [0], .symbol_desc [1], '_' )
529 0810 2 - .symbol_desc [1])) LSS 0
530 0811 2 THEN
531 0812 2 BEGIN
532 0813 2 syntax_error(.tparse, emsg (noundrsc));
533 0814 2 RETURN false;
534 0815 2 END
535 0816 2 ELSE
536 0817 2 IF .checklen NEQ .prefix_desc[0] OR
537 0818 2 CH$NEQ(.prefix_desc[0], .prefix_desc[1], .checklen, .symbol_desc[1])
538 0819 2 THEN
539 0820 2 BEGIN
540 0821 2 CH$MOVE(.checklen, .symbol_desc[1], .pre_dsc[1]);
541 0822 2 pre_dsc [0] = .checklen;
542 0823 2 END;
543 0824 2
544 0825 2 symbol_desc [0] = .symbol_desc [0] - .pre_dsc [0] - 1; ! Remove underscore from constant name since
545 0826 2 symbol_desc [1] = .symbol_desc [1] + .pre_dsc [0] + 1; ! MUST have an underscore even for a nul
546 0827 2 END;
547 0828 2
```

```

: 548      0829 2 IF .comment_const
: 549      0830 2 THEN
: 550      0831 2 BEGIN
: 551      0832 2     delim_desc [0] = 1;
: 552      0833 2     delim_desc [1] = .blank [1];
: 553      0834 2     CH$FILL( ' ', con_offset, buffer);
: 554      0835 2     buffer_desc [0] = con_offset;
: 555      0836 2     CONCAT( buffer_desc, const );
: 556      0837 2     sdl_put_record( buffer_desc );
: 557      0838 2     comment_const = false;
: 558      0839 2 END;
: 559      0840 2
: 560      0841 2 buffer_desc [0] = cname_offset;
: 561      0842 2
: 562      0843 2 CH$FILL ( ' ', pre_offset, buffer );
: 563      0844 2
: 564      P 0845 2 CONCAT ( buffer_desc,
: 565      P 0846 2     delim_desc,
: 566      P 0847 2     dbl_quote,
: 567      P 0848 2     symbol_desc,
: 568      0849 2     dbl_quote );
: 569      0850 2
: 570      0851 2 buffer_desc [0] = MAX ( equals_offset, .buffer_desc[0]);
: 571      0852 2
: 572      0853 2 IF ..value LEQU dectohex_switch
: 573      0854 2 THEN
: 574      0855 2     (CONCAT (buffer_desc, ul ))
: 575      0856 2
: 576      0857 2 ELSE
: 577      0858 2     (CONCAT (buffer_desc, xl ));
: 578      0859 2
: 579      0860 2 buffer_desc [0] = MAX ( pre_offset, .buffer_desc[0]) ;
: 580      0861 2
: 581      P 0862 2 CONCAT ( buffer_desc,
: 582      P 0863 2     pre,
: 583      P 0864 2     dbl_quote,
: 584      P 0865 2     pre_desc,
: 585      P 0866 2     dbl_quote,
: 586      0867 2     tag );
: 587      0868 2
: 588      0869 2 faoout_desc [0] = line_size ;
: 589      0870 2 faoout_desc [1] = faoout_buf;
: 590      0871 2
: 591      P 0872 2 $FAO ( buffer_desc, faoout_desc [0],
: 592      0873 2     faoout_desc, ..value );
: 593      0874 2
: 594      0875 2 sdl_put_record( faoout_desc );
: 595      0876 2
: 596      0877 2 RETURN true;
: 597      0878 2
: 598      0879 2 1 END;
```

! Tack on comment if there is one...

! Start setting up output buffer

! Put sdl constant together

.EXTRN MSG\$_NOFACIL, MSG\$_NOUNDRSC

01FC 00000

.ENTRY SDL_DEFINE_CONSTANT, Save R2,R3,R4,R5,R6,- : 0715

		58	0000'	CF	9E	00002	MOVA	R7,R8		
		5E	FED0	CE	9E	00007	MCVAB	CONST, R8		
		11	0000'	CF	E8	0000C	BLBS	-304(SP), SP		0770
			00000000G	8F	DD	00011	PUSHL	MOD INIT, 1\$		0773
			10	AC	DD	00017	PUSHL	#MSG\$ NOFACIL		
	0000G	CF		02	FB	0001A	CALLS	TPARSE		
				01ED	31	0001F	BRW	#2, SYNTAX_ERROR		0774
		50	04	AC	DD	00022	1\$:	12\$		0780
		6E		60	7D	00026	MOVQ	NAME_DESC, R0		
	FF68	CD	FF6C	CD	9E	00029	MOVQ	(R0), SYMBOL_DESC		
	F0	AD		01	DD	00030	MOVAB	BUFFER, BUFFER_DESC+4		0782
	F4	AD	3C	A8	DD	00034	MOVL	#1, DELIM_DESC		0787
		52	0000'	CF	DD	00039	MOVL	COMMA+4, DELIM_DESC+4		0788
	F8	AD		52	DD	0003E	MOVL	PREFIX_DESC, R2		0793
	FC	AD	0000'	CF	DD	00042	MOVL	R2, PRE_DSC		
		50	0C	AC	E8	00048	MOVL	PREFIX_DESC+4, PRE_DSC+4		0794
04	BE	6E	5F	8F	3A	0004C	BLBS	MSG_FLAG, 6\$		0799
				02	12	00052	LOCC	#95, SYMBOL_DESC, @SYMBOL_DESC+4		0809
				51	D4	00054	BNEQ	2\$		
	56	51	04	AE	C3	00056	2\$:	R1		
				11	18	0005B	SUBL3	SYMBOL_DESC+4, R1, CHECKLEN		0810
			00000000G	8F	DD	0005D	BGEQ	3\$		
			10	AC	DD	00063	PUSHL	#MSG\$ NOUNDRSC		0813
	0000G	CF		02	FB	00066	PUSHL	TPARSE		
				01A5	31	0006B	CALLS	#2, SYNTAX_ERROR		
		52		56	D1	0006E	3\$:	13\$		0814
				0B	12	00071	BRW	CHECKLEN, R2		0817
56	00	0000'	DF	52	2D	00073	CMP	4\$		
				04	BE	0007A	BNEQ	R2, @PREFIX_DESC+4, #0, CHECKLEN, -		0818
				0A	13	0007C	CMPC5	@SYMBOL_DESC+4		
	FC	BD	04	BE	56	0007E	4\$:	5\$		
		F8	AD	56	DD	00084	BEQL	CHECKLEN, @SYMBOL_DESC+4, @PRE_DSC+4		0821
	50		6E	F8	AD	C3	00088	5\$:		0822
		6E	FF	AD	9E	0008D	MOVQ	CHECKLEN, PRE_DSC		0825
	50	04	AE	F8	AD	C1	00091	SUBL3	PRE_DSC, SYMBOL_DESC, R0	
		04	AE	01	AD	9E	00097	MOVAB	-1(R0), SYMBOL_DESC	
		36	0000'	CF	E9	0009C	6\$:	ADDL3	PRE_DSC, SYMBOL_DESC+4, R0	0826
		F0	AD	01	DD	000A1	MOVAB	1(R0), SYMBOL_DESC+4		
		F4	AD	CC	A8	DD	000A5	BLBC	COMMENT CONST, 7\$	0829
	FF6C	CD	20202020	8F	DD	000AA	MOVL	#1, DELIM_DESC		0832
	0094	CE		04	DD	000B3	MOVL	BLANK+4, DELIM_DESC+4		0833
	50	0094	CE	FF68	CD	C1	000B8	MOVL	#538976288, BUFFER	0834
	60	04	B8	68	28	000C0	MOVL	#4, BUFFER_DESC		0835
		0094	CE	68	DD	000C5	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0		0836
			0094	CE	9F	000CA	MOVQ	CONST, @CONST+4, (R0)		
		0000V	CF	01	FB	000CE	ADDL2	CONST, BUFFER_DESC		
				07	DD	000D7	7\$:	PUSHAB	BUFFER_DESC	0837
		0094	CE	00	2C	000DC	CALLS	#1, SDC_PUT_RECORD		0838
28	20			00	2C	000DC	CLRB	COMMENT CONST		0841
				CD		000E1	MOVL	#7, BUFFER_DESC		0843
	50	0094	CE	FF6C	CD	C1	000E4	MOVQ	#0, (SP), #32, #40, BUFFER	
	60	F4	BD	F0	AD	28	000EC	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	0849
		0094	CE	F0	AD	C0	000F2	MOVQ	DELIM_DESC, @DELIM_DESC+4, (R0)	
			56	D4	A8	7D	000F8	ADDL2	DELIM_DESC, BUFFER_DESC	
	50	0094	CE	FF68	CD	C1	000FC	MOVL	DBL_QUOTE, R6	
	60		67	56	28	00104	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0		
							MOVQ	R6, (R7), (R0)		

50	0094	CE		56	C0	00108	ADDL2	R6, BUFFER_DESC	
60	0094	CE	FF68	CD	C1	0010D	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	04	BE		6E	28	00115	MOVC3	SYMBOL_DESC, @SYMBOL_DESC+4, (R0)	
50	0094	CE		6E	C0	0011A	ADDL2	SYMBOL_DESC, BUFFER_DESC	
60	0094	CE	FF68	CD	C1	0011F	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	67			56	28	00127	MOVC3	R6, (R7), (R0)	
	0094	CE		56	C0	0012B	ADDL2	R6, BUFFER_DESC	
	50		0094	CE	D0	00130	MOVL	BUFFER_DESC, R0	0851
	16			50	D1	00135	CMPL	R0, #22	
				03	18	00138	BGEQ	8\$	
				16	D0	0013A	MOVL	#22, R0	
50	0094	CE		50	D0	0013D	MOVL	R0, BUFFER_DESC	
	0094	CE	FF68	CD	C1	00142	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	0855
	0000FFFF	8F	08	BC	D1	0014A	CMPL	@VALUE, #65535	0853
				11	1A	00152	BGTRU	9\$	
60	0000'	DF	0000'	CF	28	00154	MOVC3	UL, @UL+4, (R0)	0855
	0094	CE	0000'	CF	C0	0015C	ADDL2	UL, BUFFER_DESC	
				0F	11	00163	BRB	10\$	0853
60	0000'	DF	0000'	CF	28	00165	MOVC3	XL, @XL+4, (R0)	0858
	0094	CE	0000'	CF	C0	0016D	ADDL2	XL, BUFFER_DESC	
			50	CE	D0	00174	MOVL	BUFFER_DESC, R0	0860
			28	50	D1	00179	CMPL	R0, #40	
				03	18	0017C	BGEQ	11\$	
				28	D0	0017E	MOVL	#40, R0	
	0094	CE		50	D0	00181	MOVL	R0, BUFFER_DESC	
50	0094	CE	FF68	CD	C1	00186	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	0867
60	14	B8	10	A8	28	0018E	MOVC3	PRE, @PRE+4, (R0)	
	0094	CE	10	A8	C0	00194	ADDL2	PRE, BUFFER_DESC	
50	0094	CE	FF68	CD	C1	0019A	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
60	0094	CE		56	28	001A2	MOVC3	R6, (R7), (R0)	
	0094	CE		56	C0	001A6	ADDL2	R6, BUFFER_DESC	
50	0094	CE	FF68	CD	C1	001AB	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
60	FC	BD	F8	AD	28	001B3	MOVC3	PRE_DSC, @PRE_DSC+4, (R0)	
	0094	CE	F8	AD	C0	001B9	ADDL2	PRE_DSC, BUFFER_DESC	
50	0094	CE	FF68	CD	C1	001BF	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
60	0094	CE		56	28	001C7	MOVC3	R6, (R7), (R0)	
	0094	CE		56	C0	001CB	ADDL2	R6, BUFFER_DESC	
50	0094	CE	FF68	CD	C1	001D0	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
60	30	B8	2C	A8	28	001D8	MOVC3	TAG, @TAG+4, (R0)	
	0094	CE	2C	A8	C0	001DE	ADDL2	TAG, BUFFER_DESC	
	008C	CE	84	8F	9A	001E4	MOVZBL	#132, FAOUT_DESC	0869
	0090	CE	08	AE	9E	001EA	MOVAB	FAOUT_BUF, FAOUT_DESC+4	0870
			08	BC	DD	001F0	PUSHL	@VALUE	0873
			0090	CE	9F	001F3	PUSHAB	FAOUT_DESC	
			0094	CE	9F	001F7	PUSHAB	FAOUT_DESC	
			FF64	CD	9F	001FB	PUSHAB	BUFFER_DESC	
	00000000G	00		04	FB	001FF	CALLS	#4, SYSSFAO	
			008C	CE	9F	00206	PUSHAB	FAOUT_DESC	0875
	0000V	CF		01	FB	0020A	CALLS	#1, SDC_PUT_RECORD	
		50		01	D0	0020F	MOVL	#1, R0	0877
				04		00212	RET		
				50	D4	00213	CLRL	R0	0879
				04		00215	RET		

```

: 600      0880 1 GLOBAL ROUTINE sdl_end_mod =
: 601      0881 1
: 602      0882 1 |++
: 603      0883 1 |
: 604      0884 1 |
: 605      0885 1 |       This routine outputs a semicolon and an END_MODULE statement
: 606      0886 1 |       to the sdl file to end the constant definitions for a
: 607      0887 1 |       facility. Note that if the flag comment_const is true,
: 608      0888 1 |       a semicolon has already delimited a constant string, so
: 609      0889 1 |       simply output END_MODULE. This routine also resets
: 610      0890 1 |       flags to prepare for more facility definitions.
: 611      0891 1 |
: 612      0892 1 |       Inputs:
: 613      0893 1 |
: 614      0894 1 |       none
: 615      0895 1 |
: 616      0896 1 |       Outputs:
: 617      0897 1 |
: 618      0898 1 |       none
: 619      0899 1 |
: 620      0900 1 |--
: 621      0901 1 |
: 622      0902 2 BEGIN
: 623      0903 2
: 624      0904 2 LOCAL
: 625      0905 2     buffer      :      VECTOR [line_size, BYTE],
: 626      0906 2     buffer_desc :      VECTOR [2];
: 627      0907 2 |
: 628      0908 2 |     If this is the first facility, simply return.
: 629      0909 2 |
: 630      0910 2 IF NOT .mod_init
: 631      0911 2 THEN
: 632      0912 2     RETURN true;
: 633      0913 2 |
: 634      0914 2 |
: 635      0915 2 |     Reset flags to prepare for a new module
: 636      0916 2 |
: 637      0917 2 mod_init = false;      ! Currently no active modules,
: 638      0918 2 current_offset = 0;    ! restore indentation level for sdl output,
: 639      0919 2 scbs_saved = false;  ! there are no saved SDL comment blocks.
: 640      0920 2 |
: 641      0921 2 buffer_desc [1] = buffer;
: 642      0922 2 |
: 643      0923 2 CH$FILL ( ' ', cname_offset, buffer );
: 644      0924 2 |
: 645      0925 2 |
: 646      0926 2 |     If the facility didn't end with comments, we need to add a semicolon
: 647      0927 2 |     to delimit the constants
: 648      0928 2 |
: 649      0929 2 IF NOT .comment_const
: 650      0930 2 THEN
: 651      0931 3 BEGIN
: 652      0932 3     buffer_desc [0] = cname_offset;
: 653      0933 3     CONCAT ( buffer_desc, semicolon );
: 654      0934 3     sdl_put_record ( buffer_desc );
: 655      0935 2 END;
: 656      0936 2
```

```
: 657      0937 2 1
: 658      0938 2 1 Now add the END_MODULE
: 659      0939 2 1
: 660      0940 2 buffer_desc [0] = mod_offset ;
: 661      0941 2 CONCAT ( buffer_desc, endmodule );
: 662      0942 2 sdl_put_record ( buffer_desc );
: 663      0943 2 sdl_put_record ( UPLIT ( DESC ( ' ' ) ) );
: 664      0944 2 1
: 665      0945 2 1 Reset the comment_const flag now that we're done with it
: 666      0946 2 1
: 667      0947 2 1
: 668      0948 2 comment_const = false;
: 669      0949 2 1
: 670      0950 2 RETURN true;
: 671      0951 2 1
: 672      0952 1 END;
```

.PSECT \$PLITS\$,NOWRT,NOEXE,2

```
20 0012C P.ABJ: .ASCII \ \
00000001 0012D .BLKB 3
00000000' 00130 P.ABI: .LONG 1
00000000' 00134 .ADDRESS P.ABJ
```

.PSECT \$CODE\$,NOWRT,2

07	20	04	AE	01FC	00000	.ENTRY	SDL_END_MOD, Save R2,R3,R4,R5,R6,R7,R8	0880	
			58	0000V	CF 9E 00002	MOVAB	SDL_PUT_RECORD, R8		
			57	0000'	CF 9E 00007	MOVAB	MOD_INIT, R7		
			56	0000'	CF 9E 0000C	MOVAB	SEMICOLON, R6		
			5E	FF74	CE 9E 00011	MOVAB	-140(SP), SP		
			4E		67 E9 00016	BLBC	MOD_INIT, 2\$	0910	
					67 94 00019	CLRB	MOD_INIT	0917	
				FC	A7 D4 0001B	CLRL	CURRENT_OFFSET	0918	
				F8	A7 94 0001E	CLRB	SCBS_SAVED	0919	
			08	AE	9E 00021	MOVAB	BUFFER, BUFFER_DESC+4	0921	
			6E	00	2C 00026	MOVC5	#0, (SP), #32, #7, BUFFER	0923	
				08	AE	0002B			
			15	01	A7 E8 0002D	BLBS	COMMENT_CONST, 1\$	0929	
			6E		07 D0 00031	MOVL	#7, BUFFER_DESC	0932	
	50		6E	04	AE C1 00034	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	0933	
	60	04	B6		66 28 00039	MOVC3	SEMICOLON, @SEMICOLON+4, (R0)		
			6E		66 C0 0003E	ADDL2	SEMICOLON, BUFFER_DESC		
					5E DD 00041	PUSHL	SP	0934	
			68		01 FB 00043	CALLS	#1, SDL_PUT_RECORD		
			6E		01 D0 00046	MOVL	#1, BUFFER_DESC	0940	
	50		6E	04	AE C1 00049	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	0941	
	60	18	B6	14	A6 28 0004E	MOVC3	ENDMODULE, @ENDMODULE+4, (R0)		
			6E	14	A6 C0 00054	ADDL2	ENDMODULE, BUFFER_DESC		
					5E DD 00058	PUSHL	SP	0942	
			68		01 FB 0005A	CALLS	#1, SDL_PUT_RECORD		
				0098	C6 9F 0005D	PUSHAB	P.ABI	0943	
			68		01 FB 00061	CALLS	#1, SDL_PUT_RECORD		

SDLGEN
V04-000

K 16
16-Sep-1984 02:11:38
14-Sep-1984 12:46:23

VAX-11 Bliss-32 V4.0-742
[MSGFIL.SRC]SDLGEN.B32;1

Page 22
(4)

50 01 A7 94 00064 CLR3 COMMENT_CONST
 01 D0 00067 2\$: MOVL #1, R0
 04 0006A RET

: 0948
: 0950
: 0952

; Routine Size: 107 bytes, Routine Base: \$CODE\$ + 04E8

```
: 674      0953 1 GLOBAL ROUTINE sdl_comment ( comment_desc, new_line_flag ) =
: 675      0954 1
: 676      0955 1 !++
: 677      0956 1
: 678      0957 1
: 679      0958 1      This routine outputs a line of comment into the SDL file.
: 680      0959 1      If the current comment comes between two SDL constants,
: 681      0960 1      the constant list must be ended for the comment and
: 682      0961 1      resumed when there are no more comments.
: 683      0962 1
: 684      0963 1
: 685      0964 1      Inputs:
: 686      0965 1
: 687      0966 1          comment_desc:
: 688      0967 1              descriptor for comment line
: 689      0968 1
: 690      0969 1          new_line_flag:
: 691      0970 1              set if comment should be placed on a new line
: 692      0971 1
: 693      0972 1      Outputs:
: 694      0973 1
: 695      0974 1          none
: 696      0975 1
: 697      0976 1      --
: 698      0977 1
: 699      0978 2 BEGIN
: 700      0979 2
: 701      0980 2 MAP
: 702      0981 2     comment_desc      : REF VECTOR;
: 703      0982 2
: 704      0983 2 LOCAL
: 705      0984 2     buffer          : VECTOR [line_size, BYTE],
: 706      0985 2     buffer_desc      : VECTOR [2];
: 707      0986 2
: 708      0987 2 buffer_desc [1] = buffer;
: 709      0988 2
: 710      0989 2     Can't intermix comments and constant expressions in SDL:
: 711      0990 2     if the variable "current_offset" has a value > 4 then
: 712      0991 2     sdlgen is in the middle of defining constants.
: 713      0992 2
: 714      0993 2
: 715      0994 3 IF (.current_offset GTR 4) AND (NOT .comment_const)
: 716      0995 2 THEN
: 717      0996 3 BEGIN
: 718      0997 3     CH$FILL( ' ', .current_offset, buffer);
: 719      0998 3     buffer_desc [0] = .current_offset;
: 720      0999 3     CONCAT buffer_desc, semicolon );
: 721      1000 3     comment_const = true;
: 722      1001 3     sdl_put_record (buffer_desc);
: 723      1002 2     END;
: 724      1003 2
: 725      1004 2 buffer_desc [0] = 0;
: 726      1005 2
: 727      1006 2 IF NOT .new_line_flag
: 728      1007 2 THEN
: 729      1008 3 BEGIN
: 730      1009 3     buffer_desc [0] = comment_skip;
```

```
: 731      1010 3      CH$FILL ( ' ', comment_skip, buffer);
: 732      1011 2      END;
: 733      1012 2
: 734      1013 2      CONCAT ( buffer_desc, sdlcomment, .comment_desc );
: 735      1014 2
: 736      1015 2      !
: 737      1016 2      ! If the module is not yet initialized, then the comment just
: 738      1017 2      ! encountered must be saved until a module declaration can be
: 739      1018 2      ! made. Set up a linked list to save the comments.
: 740      1019 2      !
: 741      1020 2
: 742      1021 2      IF NOT .mod_init
: 743      1022 2      THEN
: 744      1023 3      BEGIN
: 745      1024 3      LOCAL
: 746      1025 3      status,
: 747      1026 3      CBLOCK: REF BLOCK[,BYTE];
: 748      1027 3
: 749      1028 3      status = LIB$GET_VM(%REF(SCB$C_LENGTH + .buffer_desc[0]), CBLOCK);
: 750      1029 3      IF .status
: 751      1030 3      THEN
: 752      1031 3      BEGIN
: 753      1032 4      ! This is the first comment encountered--set up linked list
: 754      1033 4      !
: 755      1034 4      ! IF NOT .scbs_saved
: 756      1035 4      ! THEN
: 757      1036 4      ! BEGIN
: 758      1037 4      !     scb_head = .CBLOCK;
: 759      1038 4      !     last_scb = .CBLOCK;
: 760      1039 5      !     scbs_saved = true;
: 761      1040 5      !     END
: 762      1041 5      !
: 763      1042 5      ! Else this is at least the second comment link--set the
: 764      1043 5      ! forward ptr of the last block to point here.
: 765      1044 5      !
: 766      1045 5      ! ELSE
: 767      1046 5      ! BEGIN
: 768      1047 5      !     .last_scb = .CBLOCK;      ! Setpointer of last block to this block
: 769      1048 5      !     last_scb = .CBLOCK;      ! Now point to this block
: 770      1049 4      ! END;
: 771      1050 5      !
: 772      1051 5      ! Now set the current next link to 0;
: 773      1052 5      ! fill in this scb with the current comment string
: 774      1053 4      !
: 775      1054 4      ! CBLOCK [SCB$L_NXTSCB] = 0;
: 776      1055 4      ! CBLOCK [SCB$B_CLEN] = .buffer_desc [0];
: 777      1056 4      ! CH$MOVE (.buffer_desc [0],
: 778      1057 4      !     .buffer_desc [1],
: 779      1058 4      !     CBLOCK [SCB$T_TXT] );
: 780      1059 4      !
: 781      1060 4      ! END
: 782      1061 4      !
: 783      1062 4      !
: 784      1063 4      !
: 785      1064 4      !
: 786      1065 4      !
: 787      1066 4      !
```

```

: 788      1067 4      ! If no virtual memory available in which to save comments
: 789      1068 4      ! (hopefully a rare occurrence) then signal that condition
: 790      1069 4
: 791      1070 3      ELSE
: 792      1071 3      SIGNAL ( .status );
: 793      1072 3
: 794      1073 3      END
: 795      1074 3
: 796      1075 3      !
: 797      1076 3      ! If module has been declared and we're not in the midst of a
: 798      1077 3      ! string of constants, output the comment.
: 799      1078 3
: 800      1079 2      ELSE
: 801      1080 2
: 802      1081 2      sdl_put_record ( buffer_desc,
: 803      1082 2      IF NOT .new_line_flag THEN true ELSE false );
: 804      1083 2      RETURN true;
: 805      1084 2
: 806      1085 1      END;
```

				01FC 00000	.ENTRY	SDL COMMENT, Save R2,R3,R4,R5,R6,R7,R8	: 0953
		58	0000'	CF 9E 00002	MOVAB	SEMICOLON, R8	
		57	0000'	CF 9E 00007	MOVAB	LAST_SCB, R7	
		5E	FF6C	CE 9E 0000C	MOVAB	-148(SP), SP	
	0C	AE	10	AE 9E 00011	MOVAB	BUFFER, BUFFER_DESC+4	: 0987
		56	08	A7 D0 00016	MOVL	CURRENT_OFFSET, R6	: 0994
		04		56 D1 0001A	CMPL	R6, #4	
				2A 15 0001D	BLEQ	1\$	
		26	0D	A7 E8 0001F	BLBS	COMMENT CONST, 1\$	
56	20	6E		00 2C 00023	MOVCS	#0, (SP), #32, R6, BUFFER	: 0997
			10	AE 00028			
		08	AE	56 D0 0002A	MOVL	R6, BUFFER_DESC	: 0998
	50	08	AE	0C AE C1 0002E	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	: 0999
	60	04	B8	68 28 00034	MOVCS	SEMICOLON, @SEMICOLON+4, (R0)	
		08	AE	68 C0 00039	ADDL2	SEMICOLON, BUFFER_DESC	
		0D	A7	01 90 0003D	MOVB	#1, COMMENT_CONST	: 1000
			08	AE 9F 00041	PUSHAB	BUFFER_DESC	: 1001
		0000V	CF	01 FB 00044	CALLS	#1, SDL_PUT_RECORD	
			08	AE D4 00049	CLRL	BUFFER_DESC	: 1004
		0B	08	AC E8 0004C	BLBS	NEW_LINE_FLAG, 2\$	: 1006
		08	AE	05 D0 00050	MOVL	#5, BUFFER_DESC	: 1009
05	20	6E		00 2C 00054	MOVCS	#0, (SP), #32, #5, BUFFER	: 1010
			10	AE 00059			
	50	08	AE	0C AE C1 0005B	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	: 1013
	60	34	B8	30 A8 28 00061	MOVCS	SDLCOMMENT, @SDLCOMMENT+4, (R0)	
		08	AE	30 A8 C0 00067	ADDL2	SDLCOMMENT, BUFFER_DESC	
		56		04 AC D0 0006C	MOVL	COMMENT_DESC, R6	
	50	08	AE	0C AE C1 00070	ADDL3	BUFFER_DESC+4, BUFFER_DESC, R0	
	60	04	B6	66 28 00076	MOVCS	(R6), @4(R6), (R0)	
		08	AE	66 C0 0007B	ADDL2	(R6), BUFFER_DESC	
		4D		0C A7 E8 0007F	BLBS	MOD_INIT, 6\$	: 1021
			04	AE 9F 00083	PUSHAB	CBLOCK	: 1029
04	AE	0C	AE	05 C1 00086	ADDL3	#5, BUFFER_DESC, 4(SP)	

00000000G	00	04	AE	9F	0008C	PUSHAB	4(SP)	:	
	2C		02	FB	0008F	CALLS	#2, LIB\$GET_VM	:	1030
	50	04	50	E9	00096	BLBC	STATUS, 5\$	:	1040
	0D	04	AE	D0	00099	MOVL	CBLOCK, R0	:	1037
FC	A7		A7	E8	0009D	BLBS	SCBS_SAVED, 3\$	:	1040
	67		50	D0	000A1	MOVL	R0, SCB_HEAD	:	1041
04	A7		50	D0	000A5	MOVL	R0, LAST_SCB	:	1042
			01	90	000A8	MOVB	#1, SCBS_SAVED	:	1037
00	B7		07	11	000AC	BRB	4\$	:	1051
	67		50	D0	000AE	3\$: MOVL	R0, @LAST_SCB	:	1052
			50	D0	000B2	MOVL	R0, LAST_SCB	:	1059
04	A0		60	D4	000B5	4\$: CLRL	(R0)	:	1060
05	OC	04	AE	90	000B7	MOVB	BUFFER_DESC, 4(R0)	:	1063
	BE	08	AE	28	000BC	MOVC3	BUFFER_DESC, @BUFFER_DESC+4, 5(R0)	:	1030
			1D	11	000C3	BRB	9\$	:	1071
00000000G	00		50	DD	000C5	5\$: PUSHL	STATUS	:	1021
			01	FB	000C7	CALLS	#1, LIB\$SIGNAL	:	1082
	04	08	12	11	000CE	BRB	9\$	:	
			AC	E8	000D0	6\$: BLBS	NEW_LINE_FLAG, 7\$	:	
			01	DD	000D4	PUSHL	#1	:	
			02	11	000D6	BRB	8\$	:	
			7E	D4	000D8	7\$: CLRL	-(SP)	:	
0000V	CF	0C	AE	9F	000DA	8\$: PUSHAB	BUFFER_DESC	:	1081
	50		02	FB	000DD	CALLS	#2, SDC_PUT_RECORD	:	1083
			01	D0	000E2	9\$: MOVE	#1, R0	:	1085
			04	000E5	RET			:	

; Routine Size: 230 bytes, Routine Base: \$CODE\$ + 0553

```

808 1086 1 GLOBAL ROUTINE sdl_put_record ( record_desc, append_to_buffer_flag ) =
809 1087 1
810 1088 1 !++
811 1089 1
812 1090 1 This routine outputs a record to the SDL file.
813 1091 1
814 1092 1 Inputs:
815 1093 1
816 1094 1 record_desc:
817 1095 1 descriptor of the record to be buffered
818 1096 1
819 1097 1 append_to_buffer_flag:
820 1098 1 set if buffer should be lengthened.
821 1099 1
822 1100 1
823 1101 1 Outputs:
824 1102 1
825 1103 1 none
826 1104 1
827 1105 1 !--
828 1106 1
829 1107 2 BEGIN
830 1108 2
831 1109 2 MAP
832 1110 2 record_desc : REF VECTOR;
833 1111 2
834 1112 2 BUILTIN
835 1113 2 NULLPARAMETER;
836 1114 2
837 1115 2 LOCAL
838 1116 2 status; ! status code
839 1117 2
840 1118 2 IF .sdl_fab [fab$w_ifi] EQL 0 ! if file unopened, then exit
841 1119 2 THEN
842 1120 2 RETURN true;
843 1121 2
844 1122 2 IF NULLPARAMETER(2) ! new line flag not input or null
845 1123 2 THEN
846 1124 2 BEGIN
847 1125 2 sdl_rab [rab$w_rsz] = .output_desc [0]; ! output and reset buffer
848 1126 2 sdl_rab [rab$l_rbf] = .output_desc [1];
849 1127 2
850 1128 2 status = $PUT (RAB = sdl_rab); ! output line
851 1129 2
852 1130 2 IF NOT .status
853 1131 2 THEN
854 1132 2 rms_error (emsg(writeerr), sdl_fab, sdl_rab); ! if error, report it
855 1133 2
856 1134 2 output_desc [0] = .record_desc [0];
857 1135 2 CH$MOVE ( .record_desc [0], .record_desc [1], .output_desc [1]);
858 1136 2
859 1137 2 RETURN status;
860 1138 2 END
861 1139 2
862 1140 2 ELSE
863 1141 2 BEGIN ! modify buffer by
864 1142 2 CONCAT ( output_desc, .record_desc ); ! appending new record

```

```
: 865      1143 3  RETURN true;
: 866      1144 2  END;
: 867      1145 2
: 868      1146 1 END;
```

				.EXTRN SYSSPUT		
				.ENTRY	SDL_PUT_RECORD, Save R2,R3,R4,R5,R6,R7	: 1086
				MOVAB	SDL_RAB, R7	:
				MOVAB	OUTPUT_DESC, R6	:
				SUBL2	#4, SP	:
				TSTW	SDL_FAB+2	: 1118
				BEQL	4\$	:
				MOVL	RECORD_DESC, R2	: 1135
				CMPB	(AP), #2	: 1122
				BLSSU	1\$	:
				TSTL	8(AP)	:
				BNEQ	3\$	:
				MOVW	OUTPUT_DESC, SDL_RAB+34	: 1125
				MOVL	OUTPUT_DESC+4, SDL_RAB+40	: 1126
				PUSHL	R7	: 1128
				CALLS	#1, SYSSPUT	:
				MOVL	R0, STATUS	:
				BLBS	STATUS, 2\$	: 1130
				PUSHL	R7	: 1132
				PUSHAB	SDL_FAB	:
				PUSHL	#9900242	:
				CALLS	#3, RMS_ERROR	:
				MOVL	@RECORD_DESC, OUTPUT_DESC	: 1134
				MOVC3	@RECORD_DESC, @4(R2), @OUTPUT_DESC+4	: 1135
				MOVAB	STATUS, R0	: 1137
				RET		: 1141
				ADDL3	OUTPUT_DESC+4, OUTPUT_DESC, R0	: 1142
				MOVC3	@RECORD_DESC, @4(R2), (R0)	:
				ADDL2	@RECORD_DESC, OUTPUT_DESC	:
				MOVL	#1, R0	: 1143
				RET		: 1146

: Routine Size: 110 bytes, Routine Base: \$CODE\$ + 0639

```
: 869      1147 1 END
: 870      1148 0 ELUDOM
```

.EXTRN LIB\$SIGNAL

PSECT SUMMARY

Name	Bytes	Attributes
\$OWNS	336	NOVEC, WRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)
\$PLITS	312	NOVEC, NOWRT, RD, NOEXE, NOSHR, LCL, REL, CON, NOPIC, ALIGN(2)

SDLGEN
V04-000

F 1
16-Sep-1984 02:11:38
14-Sep-1984 12:46:23

VAX-11 Bliss-32 V4.0-742
[MSGFIL.SRC]SDLGEN.B32;1

Page 29
(6)

; \$CODE\$

1703 NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Total	Symbols Loaded	----- Percent	Pages Mapped	Processing Time
;\$255\$DUA28:[SYSLIB]STARLET.L32;1	9776	22	0	581	00:01.0

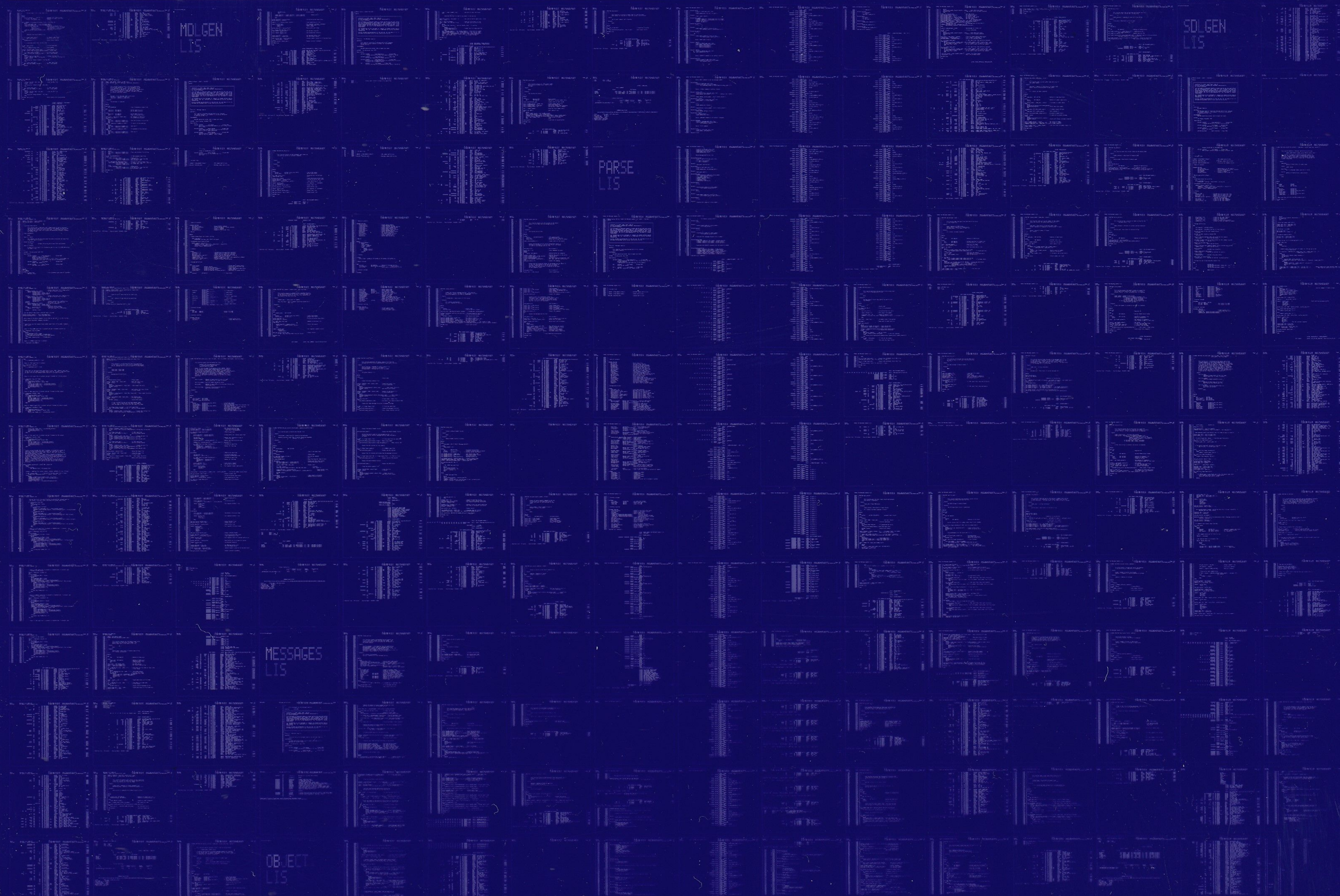
COMMAND QUALIFIERS

; BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:SDLGEN/OBJ=OBJ\$:SDLGEN MSRC\$:SDLGEN/UPDATE=(ENH\$:SDLGEN)

; Size: 1703 code + 648 data bytes
; Run Time: 00:31.0
; Elapsed Time: 01:24.3
; Lines/CPU Min: 2221
; Lexemes/CPU-Min: 28127
; Memory Used: 218 pages
; Compilation Complete

0252 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0253

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY